UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/033,242	12/28/2001	David Thomas	VIGN1340	8530

25094 7590 07/08/2004

GRAY, CARY, WARE & FREIDENRICH LLP
1221 SOUTH MOPAC EXPRESSWAY
SUITE 400
AUSTIN, TX 78746-6875

EXAMINER	
WONG, LESLIE	

ART UNIT	PAPER NUMBER
2177	3

DATE MAILED: 07/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/033,242	Applicant(s) THOMAS ET AL.	
	Examiner Leslie Wong	Art Unit 2177	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 December 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 December 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *2/03Jan2001*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

2. The abstract of the disclosure is objected to because of the following:

- It exceeds 150 words limit. Correction is required. See MPEP § 608.01(b).

- The word "comprising on line 7 incorporates legal phraseology from claim language.

3. Applicant is reminded to update the Application Serial Number and the status of the incorporated by reference application indicated on page 13 of the Specification.

The following is a quote in part of MPEP 608.01(p), concerning the incorporation of subject matter by reference:

> "The Commissioner has considerable discretion in determining what may or may not be incorporated by reference in a patent application. *General Electric Co. v. Brenner*, 407 F.2d 1258, 159 USPQ 335 (D.C. Cir. 1968). The incorporation by reference practice with reference to applications which issue as U.S. patents provides the public with a patent disclosure which minimizes the public's burden to search for and obtain copies of documents incorporated by reference which may not be readily available. Through the Office's incorporation by reference policy the Office ensures that reasonably complete disclosures are published as U.S. patents. The following is the manner in which the Commissioner has elected to exercise that discretion.
>
> An application as filed must be complete in itself in order to comply with 35 U.S.C. 112. Material nevertheless may be incorporated by reference, *Ex parte Schwarze*, 151 USPQ 426 (Bd. App. 1966). An application for a patent when filed may incorporate "essential material" by reference to (1) a U.S. patent or (2) a pending U.S. application, subject to the conditions set forth below.
>
> "Essential material" is defined as that which is necessary to (1) describe the claimed invention, (2) provide an enabling disclosure of the claimed invention, or (3) describe the best mode (35 U.S.C. 112). In any application which is to issue as a U.S. patent, essential material may not be incorporated by reference to (1) patents or applications published by foreign countries or a regional patent office, (2) non-patent publications, (3) a U.S. patent or application which itself incorporates "essential material" by reference, or (4) a foreign application.
>
> Nonessential subject matter may be incorporated by reference to (1) patents or applications published by the United States or foreign countries or regional patent offices, (2) prior filed, commonly owned U.S. applications, or (3) non-patent publications. Nonessential subject matter is subject matter referred to for purposes of indicating the background of the invention or illustrating the state of the art.
>
> Mere reference to another application, patent, or publication is not an incorporation of anything therein into the application containing such reference for the purpose of the disclosure required by 35 U.S.C. 112, first paragraph. *In re de Seversky*, 474 F.2d 671, 177 USPQ 144, (CCPA 1973). In addition to other requirements for an application, the referencing application should include an identification of the referenced patent, application, or publication. Particular attention should be directed to specific portions of the referenced document where the subject matter being incorporated may be found. Guidelines for situations where applicant is permitted to fill in a number for Serial No. _____ left blank in the application as filed can be found in *In re Fouche*, 439 F.2d 1237, 169 USPQ 429 (CCPA 1971) (Abandoned applications less than 20 years old can be incorporated by reference to same extent as copending applications; both types are open to public upon referencing application issuing as a patent)."

4. The disclosure is objected to because of the following informalities: on page 3, paragraph 0005, line 3, the word modified appears twice (i.e., modified modified).

Appropriate correction is required.

Drawings

5. The drawings are objected to because they fail to show necessary textual labels of features or symbols in Fig. 1 as described in the specification. For example, placing a label, *"client computer"*, with element 22 of Fig. 1, would give the viewer necessary detail to fully understand this element at a glance. A descriptive textual label for each numbered element in these figures would be needed to better understand these figures without substantial analysis of the detailed specification. Any structural detail that is of sufficient importance to be described should be labeled in the drawing. Optionally, the applicant may wish to include a table next to the present figure to fulfill this requirement. See 37 CFR 1.83. 37 CFR 1.84(n)(o), recited below:

> "(n) Symbols. Graphical drawing symbols may be used for conventional elements when appropriate. The elements for which such symbols and labeled representations are used must be adequately identified in the specification. Known devices should be illustrated by symbols which have a universally recognized conventional meaning and are generally accepted in the art. Other symbols which are not universally recognized may be used, subject to approval by the Office, if they are not likely to be confused with existing conventional symbols, and if they are readily identifiable.
>
> (o) Legends. Suitable descriptive legends may be used, or may be required by the Examiner, where necessary for understanding of the drawing, subject to approval by the Office. They should contain as few words as possible."

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1-6, 9-12, 14-22, 25-34, and 37-43 are rejected under 35 U.S.C. 102(e) as being anticipated by **Pace et al.** ("**Pace**") (U.S. Patent Application 20030101223A1).

Regarding claim 1, **Pace** teaches a system for synchronizing a cached file with a database:

a). **'a computer processor'** (Fig. 1G);

b). **'a network connection device operable to establish a connection with a database'** as a session bean manages the data associated with the user's connection to the database (¶s 74 and 481 and Fig. 11);

c). **'a computer readable memory containing a cache'** as static content assets may be cached in a local (i.e., client memory) for re-use so that the asset doesn't have to be resent over the network each time it is used (¶ 341);

d). and a software program, executable to run in user space, stored on the computer readable memory and executable by the computer processor to:

1). **'send a request to the database for a database asset'** as the relational data asset may be obtained through a database query such as an SQL query (¶317);

2). **'receive a database asset directly from a database'** as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522);

3). **'store the database asset as a cached file in the cache'** as a cached may be used to store the assets on disk or in memory. Each asset stored has an asset ID (i.e., filename) associated with it (¶ 526 and 523);

4). **'determine if the cached file has been modified'** as versioning is the process of determining the current version of an asset against a cached asset (¶ 790); and

5). **'if the cached file has been modified, save the cached file directly to the database'** as the Synchronize Asset Adaptor (SAA) retrieves the synchronization information from the client environment for the respective asset. In the case of an ED or EB, the retrieval information constitutes the insertion, deletion, and updating of database records which constitute the changes that the data has undergone at client since being deployed into the client target environment for this respective asset (¶s 700, 703, 709, 827 and 958).

Regarding claim 2, **Pace** further teaches **'an operating system operable to open the cached file in an application associated with a file type for the cached**

file' as PDF viewer enables the system to open and view the PDF files and MP3 Player enables the system to open and play MP3 files (¶ 339).

Regarding claim 3, **Pace** further teaches wherein **'the software program is further executable to receive a notification from a file management system of an operating system that the cached file has been modified'** as the Caching Agent Method (CAM) notifies the proper sub-systems that a change has occurred in the asset cache (¶ 827).

Regarding claim 4, **Pace** further teaches wherein **'the software program is further executable to poll the cached file to determine if the cached file has been modified'** as the client deployment process will be given a hint as to how long the agent should wait before polling for status of the client/target deployment queue (¶s 790, 79, 534, and 827).

Regarding claim 5, **Pace** further teaches wherein **'the software program is further executable to associate the cached file with a connection, wherein the connection is associated with the database'** as a connection is established between the target node and the asset's original source node. The session bean would be required for managing the data associated with the user's connection, and possibly accessing data in the relational database (¶s 78, 481 and 482).

Regarding claim 6, **Pace** further teaches wherein **'the software program is further operable to establish the connection'** as the session bean would be required for managing the data associated with the user's connection, and possibly accessing data in the relational database (¶s 74 and 481).

Regarding claim 9, **Pace** further teaches wherein **'the software program is further executable to associate the cached file with a location in the cache'** as the Asset Descriptor Manifest (ADM) may be a data structure that associates asset IDS and offsets. The offset 1556H offsets into file containing cached asset, e.g. the boundaries of the asset fragment (¶s 536 and 538).

Regarding claim 10, **Pace** further teaches wherein '**an application accessing the cached file saves the cached file at the location in the cache associated with the cached file'** as the Asset Descriptor Manifest (ADM) may be a data structure that associates asset IDS and offsets. The offset 1556H offsets into file containing cached asset, e.g. the boundaries of the asset fragment (¶s 536 and 538).

Regarding claim 11, **Pace** further teaches wherein the **'software program is further executable to receive a database notification from a database management program that an additional user has modified the database asset'** as the client DBMS would call the CDA when the database record has changed (¶ 700).

Regarding claim 12, **Pace** further teaches wherein the **'software program is further executable to provide a notice to a first user that the additional user has modified the database asset'** as the client DBMS would call the CDA when the database record has changed (¶ 700).

Regarding claim 14, **Pace** further teaches wherein **'the software program is further executable to: receive the request from a first user for the database asset'** as the relational data asset may be obtained through a database query such as an SQL query (¶317).

Regarding claim 15, **Pace** further teaches where the software program is executable to: **'receive a notification that said database asset has been deleted from said database; and purge said cached file from said cache'** as the client DBMS would call the CDA when the database record has changed and the Caching Agent Method (CAM) deletes old assets from the asset cache (¶s 700 and 827).

Regarding claim 16, **Pace** teaches a system for synchronizing a file in a cache comprising:

a database server further comprising:

'a server processor' (Fig. 11, element 1127);

'a server memory' (Fig. 11, element 1127);

'a database stored on said server memory containing a plurality of databases' as EIS tier with Web server, application server, and database server (Fig. 11); and

a database management program stored on the server memory executable by the server processor to:

'receive a client request for a database asset from the plurality of database assets' as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522); and

'retrieve the database asset' as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522);

'a client computer in electrical communication with the database server' as Internet client accesses EIS tier via network 1110 (Fig. 11) further comprising:

a client processor (Fig. 11, element 1154);

a client memory (Fig. 11, element 1154); and

'a cache manager program, stored on the client memory executable by the client processor to run in user space' as static content assets may be cached in a local (i.e., client memory). The Caching agent method (CAM) notifies the proper sub-systems that a change has occurred in the asset cache (¶s 341, 827) and to:

'establish a connection to the database server' as a session bean manages the data associated with the user's connection to the database (¶s 74 and 481);

'forward the client request for the database asset to the database server' as if the Computational Agent (CA) does not have an asset to fulfill the request, it calls the subscription agent, in order to forward the request to another CA. In the case that requests reach the CA in the source environment, the requests are made directly against the resources in that environment (¶ 818)

'receive the database asset directly from the database server' as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522);

'store the database asset as a cached file in the client memory' as a cached may be used to store the assets on disk or in memory. Each asset stored has an asset ID (i.e., filename) associated with it (¶ 526 and 523);

'determine if the cached file has been modified' as versioning is the process of determining the current version of an asset against a cached asset (¶ 790); and

'if the cached file has been modified, communicate the cached file directly to the database' as the Synchronize Asset

Adaptor (SAA) retrieves the synchronization information from the client environment for the respective asset. In the case of an ED or EB, the retrieval information constitutes the insertion, deletion, and updating of database records which constitute the changes that the data has undergone at client since being deployed into the client target environment for this respective asset (¶s 700, 703, 709, 827 and 958).

Regarding claim 17, **Pace** further teaches wherein the client computer further comprises:

'an operating system' as a transactional operating system 106 (Fig. 1A and ¶42); and

'an application executable to access the cached file' as the Caching Agent Method (CAM) performs the functionality that is required to store asset, and manage the cache, including: deleting old assets, updating database tables, and notifying the proper subsystems that a chance has occurred in the asset cache (¶ 801-803, and 827).

Regarding claim 18, **Pace** further teaches wherein **'the application is associated with a file type corresponding to the database asset'** as PDF viewer enables the system to open and view the PDF files and MP3 Player enables the system to open and play MP3 files (¶ 339).

Regarding claim 19, **Pace** further teaches wherein **'the cache manager program is further executable to prompt the operating system to access the cached file using the application'** as the adjustment asset adapter method may add a query entry that will prompt various agents for the type of information that the adjustment asset adapter method needs in order to adjust the distribution of assets (¶ 739);

Regarding claim 20, **Pace** further teaches **wherein 'the client computer further comprises: an operating system having a file management system; and wherein the cache manager program is further executable to receive a notification from the file management system that the cached file has been modified'** as the Caching agent method (CAM) notifies the proper sub-systems that a change has occurred in the asset cache (¶ 827).

Regarding claim 21, **Pace** further teaches wherein **'the cache management program is further executable to poll the cached file to determine if the cached file has been modified'** as versioning is the process of determining the current version of an asset against a cached asset. The client deployment process will be given a hint as to how long the agent should wait before polling for status of the client/target deployment queue (¶s 790, 79, 534, and 827).

Regarding claim 22, **Pace** further teaches wherein **'the cache management program is further executable to associate the cached file with a connection, wherein the connection is associated with the database'** as a connection is established between the target node and the asset's original source node. The session bean would be required for managing the data associated with the user's connection, and possibly accessing data in the relational database (¶s 78, 481 and 482).

Regarding claim 25, **Pace** further teaches wherein **'the cache management program is further executable to associate the cached file with a location in the cache'** as the Asset Descriptor Manifest (ADM) may be a data structure that associates asset IDS and offsets. The offset 1556H offsets into file containing cached asset, e.g. the boundaries of the asset fragment (¶s 536 and 538).

Regarding claim 26, **Pace** further teaches wherein **'an application accessing the cached file saves the cached file at the location in the cache associated with the cached file'** as the Asset Descriptor Manifest (ADM) may be a data structure that associates asset IDS and offsets. The offset 1556H offsets into file containing cached asset, e.g. the boundaries of the asset fragment (¶s 536 and 538).

Regarding claim 27, **Pace** further teaches wherein **'the database management program is executable to notify the client computer if an additional client modifies the database asset, and wherein the cache manager program is executable to**

receive the notification from the database management program' as the client DBMS would call the CDA when the database record has changed (¶ 700).

Regarding claim 28, **Pace** further teaches wherein **'the cache management program is further executable to provide a warning to a first user that the additional user has modified the database asset'** as the client DBMS would call the CDA when the database record has changed (¶ 700).

Regarding claim 29, **Pace** further teaches wherein **'the database management program is executable to notify the client computer that the database asset has been deleted from the database, and wherein the cache manager is operable to purged the cached file from the cache' modified'** as the client DBMS would call the CDA when the database record has changed and the Caching Agent Method (CAM) deletes old assets from the asset cache (¶s 700 and 827).

Regarding claim 30, **Pace** teaches a method for synchronizing a file in a cache comprising:

a). **'receiving a database asset directly from a database'** as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522);

b). **'storing the database asset in a cache as a cached file'** as a cached may be used to store the assets on disk or in memory. Each asset stored has an asset ID (i.e., filename) associated with it (¶ 526 and 523);

c). **'determining if the cached file has been modified'** as versioning is the process of determining the current version of an asset against a cached asset (¶ 790); and

d). **'if the cached file has been modified, communicating the cached file directly to database'** as the Synchronize Asset Adaptor (SAA) retrieves the synchronization information from the client environment for the respective asset. In the case of an ED or EB, the retrieval information constitutes the insertion, deletion, and updating of database records which constitute the changes that the data has undergone at client since being deployed into the client target environment for this respective asset (¶s 700, 703, 709, 827 and 958).

Regarding claim 31, **Pace** further teaches wherein the step of **'determining if the cached file has been modified further comprises receiving a notification from a file management system that the cached file has been modified'** as the Caching agent method (CAM) notifies the proper sub-systems that a change has occurred in the asset cache (¶ 827).

Regarding claim 32, **Pace** further teaches wherein the step of **'determining if the cached file has been modified further comprises polling the cached file'** as versioning is the process of determining the current version of an asset against a cached asset. The client deployment process will be given a hint as to how long the agent should wait before polling for status of the client/target deployment queue (¶s 790, 79, 534, and 827).

Regarding claim 33, **Pace** further teaches **'associating the cached file with a connection'** as a connection is established between the target node and the asset's original source node. The session bean would be required for managing the data associated with the user's connection, and possibly accessing data in the relational database (¶s 78, 481 and 482).

Regarding claim 34, **Pace** further teaches **'establishing the connection with the database'** as a session bean manages the data associated with the user's connection to the database (¶s 74 and 481).

Regarding claim 37, **Pace** further teaches '**associating the cached file with a location in a memory'** as the Asset Descriptor Manifest (ADM) may be a data structure that associates asset IDS and offsets. The offset 1556H offsets into file containing cached asset, e.g. the boundaries of the asset fragment (¶s 536 and 538).

Regarding claim 39, **Pace** further teaches **'opening the cached file with an application associated with a file type associated with the cached file'** as PDF viewer enables the system to open and view the PDF files and MP3 Player enables the system to open and play MP3 files (¶ 339).

Regarding claim 40, **Pace** further teaches **'purging the cached file from the cache if the database asset is deleted from the database'** as the client DBMS would call the CDA when the database record has changed and the Caching Agent Method (CAM) deletes old assets from the asset cache (¶s 700 and 827).

Regarding claim 41, **Pace** teaches a method of managing a cache comprising:

a). **'establishing a connection with a database'** as a session bean manages the data associated with the user's connection to the database (¶s 74 and 481);

b). **'retrieving a database asset from the database'** as a SQL query could be executed to obtained the results and insert those results into the final ASSET-DESCRIPTOR table (¶522);

c). **'storing the database asset in a cache as a cached file'** as a cached may be used to store the assets on disk or in memory. Each asset stored has an asset ID (i.e., filename) associated with it (¶ 526 and 523);

d). **'associating the cached file with the connection'** as a connection is established between the target node and the asset's original source node. The session

bean would be required for managing the data associated with the user's connection, and possibly accessing data in the relational database (¶s 78, 481 and 482).

Regarding claim 42, **Pace** further teaches **'receiving a notification from a file management system that the cached file has been modified'** as the client DBMS would call the CDA when the database record has changed (¶ 700).

Regarding claim 43, **Pace** further teaches wherein the step of **'determining if the cached file has been modified further comprises polling the cached file'** as versioning is the process of determining the current version of an asset against a cached asset. The client deployment process will be given a hint as to how long the agent should wait before polling for status of the client/target deployment queue (¶s 790, 79, 534, and 827).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

9. Claims 7-8, 23-24, and 35-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Pace et al.** ("**Pace**") (U.S. Patent Application 20030101223A1) as applied to claims 1-6, 9-12, 14-22, 25-34, 37, and 39-43 above in view of **Goldberg et al.** ("**Goldberg**") (U.S. Patent 6,434,543 B1).

Regarding claim 7, **Pace** does not explicitly teach wherein the software program is further executable to: determine if the connection has been disconnected; and if the connection has been disconnected, to reestablish the connection.

Goldberg, however, teaches **'determine if the connection has been disconnected; and if the connection has been disconnected, to reestablish the connection'** as when a query is complete and the connection between the client and server is released. Subsequently, when the client requests a connection to the database, the system re-establish the connection by examine connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein (col. 2, lines 37-39; col. 5, lines 58-59; col. 5, line 66 – col. 6, line 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Goldberg's** teaching would have allowed **Pace's** to allow regain access to the database by examining the connection cache to determine whether a database connection for the corresponding client exists in order to efficiently utilize the database and reduce overhead associated with establishing a database connection as suggested by **Goldberg** at col. 1, lines 7-9.

Regarding claim 8, **Pace** does not explicitly teach wherein the software program is further executable to: save a user login; and reestablish the connection using the user login.

Goldberg, however, teaches '**save a user login; and reestablish the connection using the user login'** as connection information which can include the database name, user name and login password are stored in the connection manager for each open connection. In response to a server makes a request to again connect to database, connection manager examines connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein. If it is, the associated connection handle is returned. If no existing connection is stored in the cache, then new connection is opened and stored in the cache (col. 2, lines 40-43; col. 5, line 66- col. 6, line 7).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because

Goldberg's teaching involves a connection manager that maintains an internal cache of database connections that have been opened (col. 5, lines 50-52) have allowed **Pace's** to eliminate the overhead necessary to establish the connection by reuse the same connection over and over as the server always logs on under the same database name, user name and password as suggested by **Goldberg** at col. 6, lines 14-19.

Regarding claim 23, **Pace** does not explicitly teach wherein the cache management program is further executable to: determine if the connection has been disconnected; and if the connection has been disconnected, to re-establish the connection.

Goldberg, however, teaches **'determine if the connection has been disconnected; and if the connection has been disconnected, to reestablish the connection'** as when a query is complete and the connection between the client and server is released. Subsequently, when the client requests a connection to the database, the system re-establish the connection by examine connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein (col. 2, lines 37-39; col. 5, lines 58-59; col. 5, line 66 – col. 6, line 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Goldberg's** teaching would have allowed **Pace's** to allow regain access to the database by examining the connection cache to determine whether a database

connection for the corresponding client exists in order to efficiently utilize the database and reduce overhead associated with establishing a database connection as suggested by **Goldberg** at col. 1, lines 7-9.

Regarding claim 24, **Pace** does not explicitly teach wherein the cache management program is further executable to: save a user login; and reestablish the connection using the user login.

Goldberg, however, teaches '**save a user login; and reestablish the connection using the user login'** as connection information which can include the database name, user name and login password are stored in the connection manager for each open connection. In response to a server makes a request to again connect to database, connection manager examines connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein. If it is, the associated connection handle is returned. If no existing connection is stored in the cache, then new connection is opened and stored in the cache (col. 2, lines 40-43; col. 5, line 66- col. 6, line 7).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Goldberg's** teaching involves a connection manager that maintains an internal cache of database connections that have been opened (col. 5, lines 50-52) have allowed **Pace's** to eliminate the overhead necessary to establish the connection by reuse the

same connection over and over as the server always logs on under the same database name, user name and password as suggested by **Goldberg** at col. 6, lines 14-19.

Regarding claim 35, **Pace** does not explicitly teach determining if the connection with the database has become disconnected; and if the connection with the database has become disconnected, reestablishing the connection to the database.

Goldberg, however, teaches **'determine if the connection has been disconnected; and if the connection has been disconnected, to reestablish the connection'** as when a query is complete and the connection between the client and server is released. Subsequently, when the client requests a connection to the database, the system re-establish the connection by examine connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein (col. 2, lines 37-39; col. 5, lines 58-59; col. 5, line 66 – col. 6, line 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Goldberg's** teaching would have allowed **Pace's** to allow regain access to the database by examining the connection cache to determine whether a database connection for the corresponding client exists in order to efficiently utilize the database and reduce overhead associated with establishing a database connection as suggested by **Goldberg** at col. 1, lines 7-9.

Regarding claim 36, **Pace** does not explicitly teach saving a user login and using the saved user login to reestablish the connection.

Goldberg, however, teaches '**save a user login; and reestablish the connection using the user login'** as connection information which can include the database name, user name and login password are stored in the connection manager for each open connection. In response to a server makes a request to again connect to database, connection manager examines connection cache to determine whether a database connection with the corresponding client, database and password information is stored therein. If it is, the associated connection handle is returned. If no existing connection is stored in the cache, then new connection is opened and stored in the cache (col. 2, lines 40-43; col. 5, line 66- col. 6, line 7).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Goldberg's** teaching involves a connection manager that maintains an internal cache of database connections that have been opened (col. 5, lines 50-52) have allowed **Pace's** to eliminate the overhead necessary to establish the connection by reuse the same connection over and over as the server always logs on under the same database name, user name and password as suggested by **Goldberg** at col. 6, lines 14-19.

10. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Pace et al.** ("**Pace**") (U.S. Patent Application 20030101223A1) as applied to claims 1-6, 9-12,

14-22, 25-34, 37, and 39-43 above in view of **Planzt et al.** ("**Planzt**") (U.S. Patent 6,088,702).

Regarding claim 13, **Pace** does not explicitly teach wherein the software program is further executable to provide the first user an option of overriding a modification to the database asset made by the additional user.

Planzt, however, teaches **"wherein the software program is further executable to provide the first user an option of overriding a modification to the database asset made by the additional user"** as a Group Publishing System (GPS) for enhancing collaboration between and among individuals who may be separated by distance and/or time. The administrator invokes a master editor for the document, permitting a final administrative review of all edited aspects of the project, modify any menus or selections available anywhere in the GPS, and deleting any other information from the GPS database (col. 2, lines 64-66; col. 11, lines 1-42; col. 5, lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Planzt's** teaching have allowed **Pace's** to review and finalized all edited aspects of the project via the use of the master editor and be able to modify or delete any information from the database as suggested by **Planzt** at col. 11, lines 4-6 and 16-18.

11. Claim 38 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Pace et al.** ("**Pace**") (U.S. Patent Application 20030101223A1) as applied to claims 1-6, 9-12,

14-22, 25-34, 37, and 39-43 above in view of **Carley et al.** ("**Carley**") (U.S. Patent 6,701,345).

Regarding claim 38, **Pace** does not explicitly teach notifying a first user than an additional user has accessed the database asset.

Carley, however, teaches **'notifying a first user than an additional user has accessed the database asset'** as a notification is sent to the user station that initiated the concurrently executing load process when multiple users attempt to alter the same data (col. 4, lines 42-54; Fig. 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of the cited references because **Carley's** teaching have allowed **Pace's** to enable the users to coordinate their updates so that all alterations to the data are entered by monitoring connections from the users stations and determining whether another load process is being concurrently executed as suggested by **Carley** at col. 4, lines 50-54.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,687,793 B1 issued to Thomas et al. on 03 February 2004. The subject matter disclosed therein is pertinent to that of claims 1-6, 18-35, 39, 42, and 43, (e.g., resources for cache management).

U.S. Patent 5,293,619 A issued to Dean on 08 March 1994. The subject matter disclosed therein is pertinent to that of claims 14 and 38 (e.g., collaborative use of an application simultaneously by multiple users).

U.S. Patent 6,026,413 A issued to Challenger et al. on 15 February 2000. The subject matter disclosed therein is pertinent to that of claims 1-6, 18-35, 39, 42, and 43 (e.g., determining how changes affect cached objects).

U.S. Patent 6,185,608 B1 issued to Hon et al. on 6,185,608. The subject matter disclosed therein is pertinent to that of claims 1-6, 18-35, 39, 42, and 43 (e.g., synchronization of data stored in the database and cache data).

U.S. Patent 5,829,022 A issued to Watamabe et al. on 27 October 1998. The subject matter disclosed therein is pertinent to that of claims 1-6, 18-35, 39, 42, and 43 (e.g., synchronizing object cache and page cache via page status and object status).

U.S. Patent Application 20020107835 A1 published on 08 August 2002. The subject matter disclosed therein is pertinent to that of claims 1-6, 18-35, 39, 42, and 43 (e.g., informational database request is stored in cache).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Leslie Wong whose telephone number is (703) 305-3018. The examiner can normally be reached on Monday to Friday 9:30am - 6:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E Breene can be reached on (703) 305-9790. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Leslie Wong
Patent Examiner
Art Unit 2177

LW
June 24, 2004